

Advanced Database Systems

Spring 2026

Q&A Session 4

If you require this document in an alternative format, such as large print or a coloured background, please contact milos.nikolic@ed.ac.uk

1

ADMINISTRIVIA

2

Practice Worksheet 4 is now available on Learn

ADBS Exam

Date & time available at <https://exams.is.ed.ac.uk/>

Important: Bring your calculator!

Before the exam:

Check which calculators are permitted at <https://edin.ac/4b0a79H>

Ensure your device complies with exam regulations

2

SURVEYS, SURVEYS, SURVEYS...

3

Course survey – please complete

Your feedback really matters

Helps us improve the course for future students (just like past feedback shaped this one)

UG students – please complete the NSS

The School takes your feedback seriously

Coffee vouchers available at upcoming events

3

QUESTION 1

4

4

Q1: TWO-PHASE COMMIT WITH LOGGING

5

Scenario

1 coordinator, 3 participants

Communication delays

Coordinator → participants: 30 ms (simultaneous delivery)

P1 → Coord: 5 ms P2 → Coord: 10 ms P3 → Coord: 15 ms

Logging cost

Log write + flush: 10 ms

Questions

- Total time for successful commit
- Crash after YES vote
- Crash scenarios (YES vs NO, with/without presumed abort)
- Coordinator crash scenarios

5

RECAP: 2PC (NORMAL MODE)

6

Goal: Ensure all participants either commit or abort consistently

Before 2PC begins:

Transaction operations are typically already executed locally
Changes are not yet durable / visible (pending commit)

Phase 1: Prepare (Voting)

Coordinator sends **PREPARE**

Participants decide if they can commit and respond to coordinator:

YES (ready to commit)

NO (cannot commit)

Phase 2: Commit (Decision)

Coordinator sends final decision to all participants

All YES → **COMMIT**

Any No → **ABORT**

Participants confirm completion by returning **ACK**

6

PARTICIPANT LOGGING & STATES

7

If participant votes **YES**:

Writes **PREPARE** record to stable storage

Keeps locks and resources

Enters uncertain state – relinquished the right to unilaterally abort, waits for coordinator's decision

Must be able to commit after recovery

"I promise I can
commit later"

If participant votes **NO**:

Aborts immediately, undoing local changes (if needed)

Releases locks

Sends **NO** to coordinator

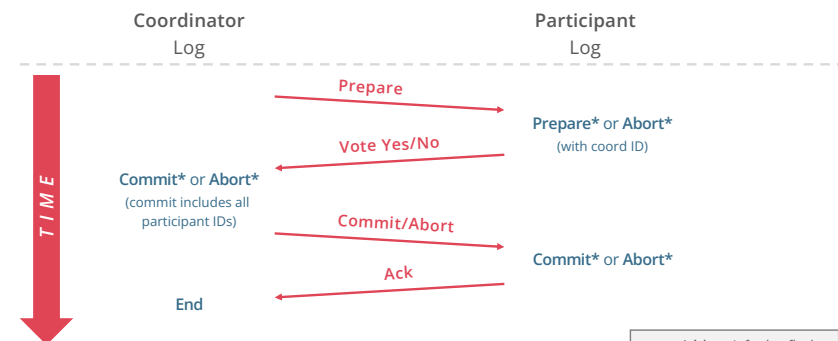
Flushes **ABORT** record

"I have already
decided to abort"

7

2PC IN A NUTSHELL

8



asterisk*: wait for log flush
before sending next msg

8

2PC OPTIMISATION: PRESUMED ABORTS

9

Core idea

If the coordinator has no record of a transaction → presume **ABORT**

What this eliminates (abort case)

Coordinator skips forced **ABORT** log write

Participants do not send **ACKs** after **ABORT**

Coordinator does no cleanup for aborted transactions (no need to write **END** records for aborts)

Why this works

Participants that vote **NO**:

Abort immediately

Typically write no durable record

Therefore: missing state safely ⇒ **ABORT**

Savings apply only to abort cases.
Commit path unchanged (still
requires logging + messages)

9

Q1A: TIMING

10

Communication delays

Coordinator → participants: **30 ms** (simultaneous delivery)

P1 → Coord: **5 ms** P2 → Coord: **10 ms** P3 → Coord: **15 ms**

Logging cost

Log write + flush: **10 ms**

Question

Under 2PC with logging, how long does the entire protocol take for a successful commit?

10

Q1A: TIMING

11

Communication delays

Coordinator → participants: **30 ms** (simultaneous delivery)

P1 → Coord: **5 ms** P2 → Coord: **10 ms** P3 → Coord: **15 ms**

Logging cost

Log write + flush: **10 ms**

Question

Under 2PC with logging, how long does the entire protocol take for a successful commit?

Phase 1: C sends **PREPARE** + Ps flush **PREPARE** + Ps send **YES** + C flushes **COMMIT**
 = $30 + 10 + \max\{5, 10, 15\} + 10 = 65$ ms

Phase 2: C sends **COMMIT** + Ps flush **COMMIT** + Ps send **ACK** + C flushes **END**
 = $30 + 10 + \max\{5, 10, 15\} + 10 = 65$ ms

Total: 130 ms

11

Q1B: FAILURE AFTER YES VOTE

12

Question

Participant

Voted **YES**

Crashes and restarts

All others voted **YES**

What happens?

12

Q1B: FAILURE AFTER YES VOTE

13

Question

Participant

Voted **YES**

Crashes and restarts

All others voted **YES**

What happens?

On recovery

Participant finds **PREPARE** record

Enters prepared (uncertain) state

Cannot decide alone → must contact coordinator

Coordinator

Cannot finish until all **ACKs** received

Knows transaction committed

Outcome

Coordinator replies **COMMIT**

Participant writes **COMMIT** record and sends **ACK**

13

Q1C.I: CRASH AFTER YES VOTE

14

Question

P1 votes **YES**, then crashes before decision

P2 votes **NO** → transaction aborts

What does P1 do after recovery

1. Without presumed abort
2. With presumed abort

14

Q1C.I: CRASH AFTER YES VOTE

15

Question

P1 votes **YES**, then crashes before decision

P2 votes **NO** → transaction aborts

What does P1 do after recovery

1. Without presumed abort
2. With presumed abort

P1 finds **PREPARE** record

Must contact coordinator

Coordinator replies: **ABORT**

P1 aborts transaction locally

Same behaviour w/ or w/o presumed abort

15

Q1C.II: CRASH AFTER NO VOTE

16

Question

P2 votes **NO**, then crashes before decision

What does P2 do after recovery

1. Without presumed abort
2. With presumed abort

16

Q1C.II: CRASH AFTER NO VOTE

17

Question

P2 votes **NO**, then crashes before decision

What does P2 do after recovery

1. Without presumed abort
2. With presumed abort

Without presumed abort

P2 has **ABORT** record log

May resend **NO** vote

Not required for correctness

Helps coordinator finish faster

Aborts locally

With presumed abort

ABORT record may be missing

No record → **presume abort**

P2 aborts locally and sends no messages

17

Q1D: COORDINATOR CRASHES EARLY

18

Scenario

Coordinator sends **PREPARE**

Crashes before receiving votes

Using presumed abort

Questions

What does coordinator do after recovery?

What does a **NO** voter do?

What does a **YES** voter do?

18

Q1D.I: COORDINATOR RECOVERY

19

Scenario

Coordinator sends **PREPARE**

Crashes before receiving votes

Using presumed abort

Questions

What does coordinator do after recovery?

What does a **NO** voter do?

What does a **YES** voter do?

19

Q1D.I: COORDINATOR RECOVERY

20

Scenario

Coordinator sends **PREPARE**

Crashes before receiving votes

Using presumed abort

Questions

What does coordinator do after recovery?

What does a **NO** voter do?

What does a **YES** voter do?

No **COMMIT** record in log

With presumed abort

No record → **ABORT**

Coordinator

Aborts locally

Writes **ABORT** record

Does not notify participants

20

Q1D.II: PARTICIPANT VOTED NO

21

Scenario

- Coordinator sends **PREPARE**
- Crashes before receiving votes
- Using presumed abort

Questions

- What does coordinator do after recovery?
- What does a **NO** voter do?
- What does a **YES** voter do?

21

Q1D.II: PARTICIPANT VOTED NO

22

Scenario

- Coordinator sends **PREPARE**
- Crashes before receiving votes
- Using presumed abort

Questions

- What does coordinator do after recovery?
- What does a **NO** voter do?
- What does a **YES** voter do?

NO vote = veto

Participant

- Already knows outcome = ABORT
- Aborts locally
- Ignores coordinator failure

22

Q1D.III: PARTICIPANT VOTED YES

23

Scenario

- Coordinator sends **PREPARE**
- Crashes before receiving votes
- Using presumed abort

Questions

- What does coordinator do after recovery?
- What does a **NO** voter do?
- What does a **YES** voter do?

23

Q1D.III: PARTICIPANT VOTED YES

24

Scenario

- Coordinator sends **PREPARE**
- Crashes before receiving votes
- Using presumed abort

Questions

- What does coordinator do after recovery?
- What does a **NO** voter do?
- What does a **YES** voter do?

Participant

- Voted **YES** → prepared state
- Cannot decide alone
- Periodically try to contact coordinator

Coordinator (after recovery)

- Returns **ABORT**

Participant

- Aborts transaction

24